

SIA TP 2

Grupo 8

Objetivo

- Crear una red neuronal capaz de estimar el valor de una función f de dominio e imagen reales
- $f(x) = \sin(x) * x^3 + x/2$, con $x \in [10, 45]$
- Implementar y comparar mejoras a la red neuronal:
 - η Adaptativo
 - *Momentum*
 - 'undo' de épocas
- Implementar y comparar cambios al dominio de la función:
 - Cambiar el intervalo de valores (min/max)
 - Modificar la cantidad de valores y el incremento
 - Normalización

Red Neuronal

- Se implementó una red neuronal utilizando Python 3.4 y *numpy*
 - Cantidades variables de capas y neuronas
 - *feed-forward*
 - *back-propagation*
 - Capa de salida de 1 neurona
 - Varias funciones de activación
- Gráficos de resultados utilizando *matplotlib*

Pruebas Iniciales

- Se probó con la función en el intervalo (10, 45, 0.1)
- Arquitectura: 20 - 5
- η : 0.1
- Función de activación: tanh
- Se entrenó durante 10000 épocas

Gráfico de función calculada

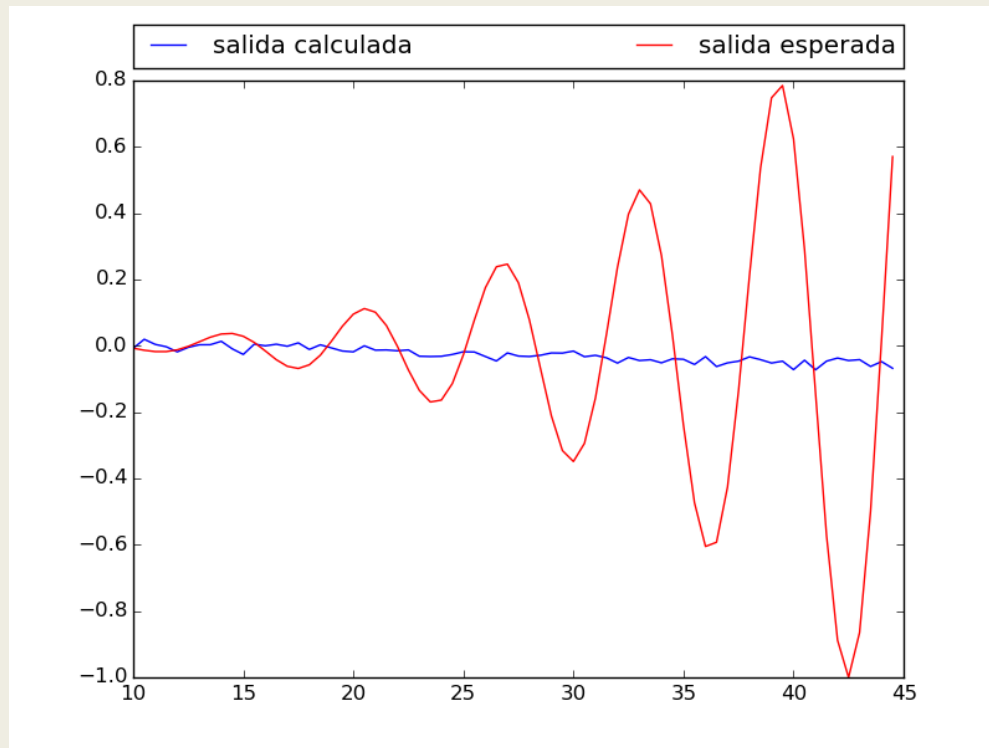
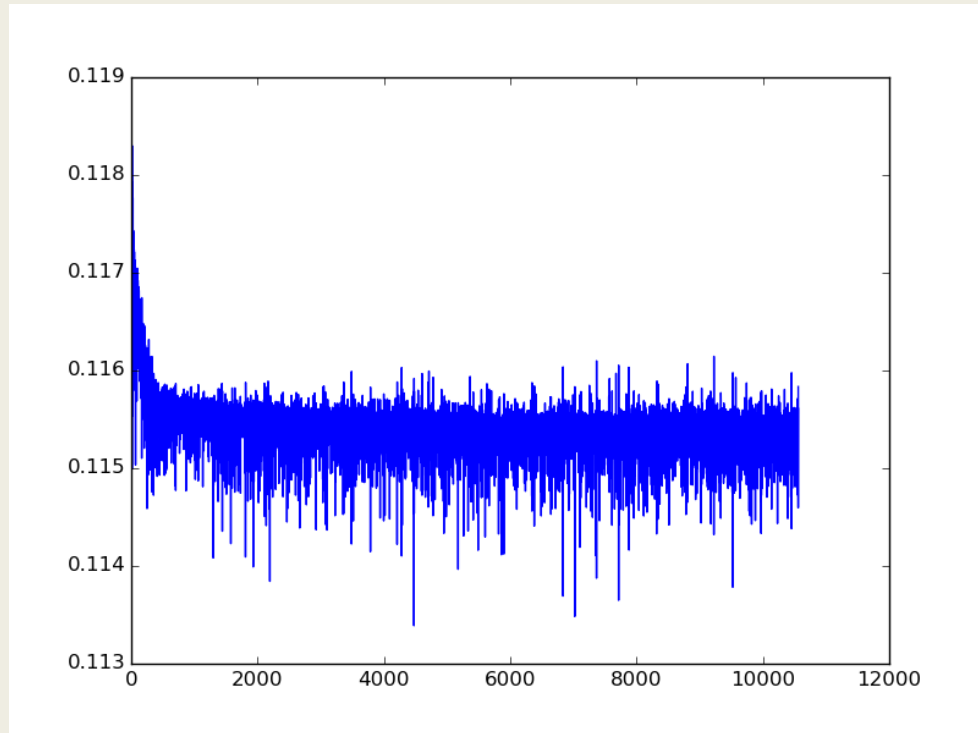


Gráfico de Error Medio



Variaciones de Arquitectura

- Se probó con una serie de distintas arquitecturas, por ejemplo:
- Variante 1:
 - 62-62-32-16-2 neuronas en las capas ocultas
 - $\eta: 3 \times 10^5$
 - g: tanh
 - rango = (10, 45, 0.5)
- Variante 2:
 - 256-128-128-64-32 neuronas en las capas ocultas
 - $\eta: 3 \times 10^5$
 - g: tanh
 - rango: (10, 45, 0.1)

Gráfico Variante 1

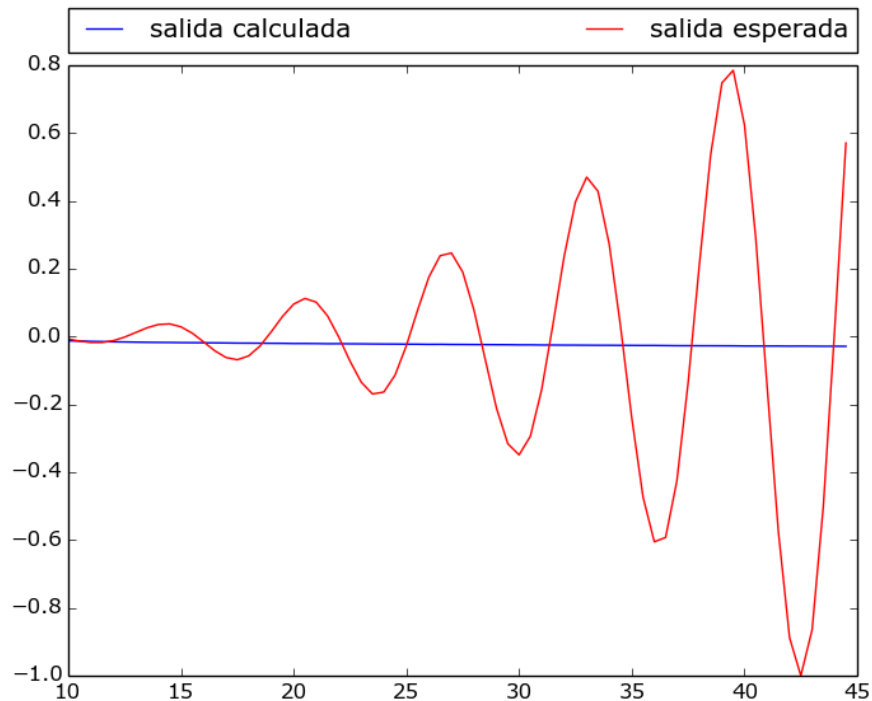
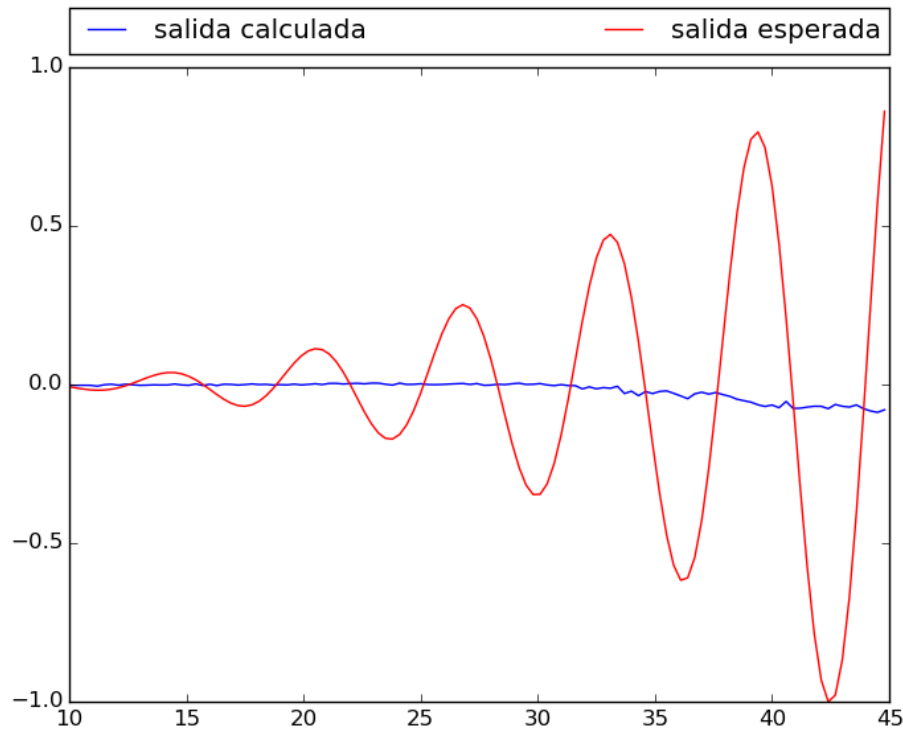


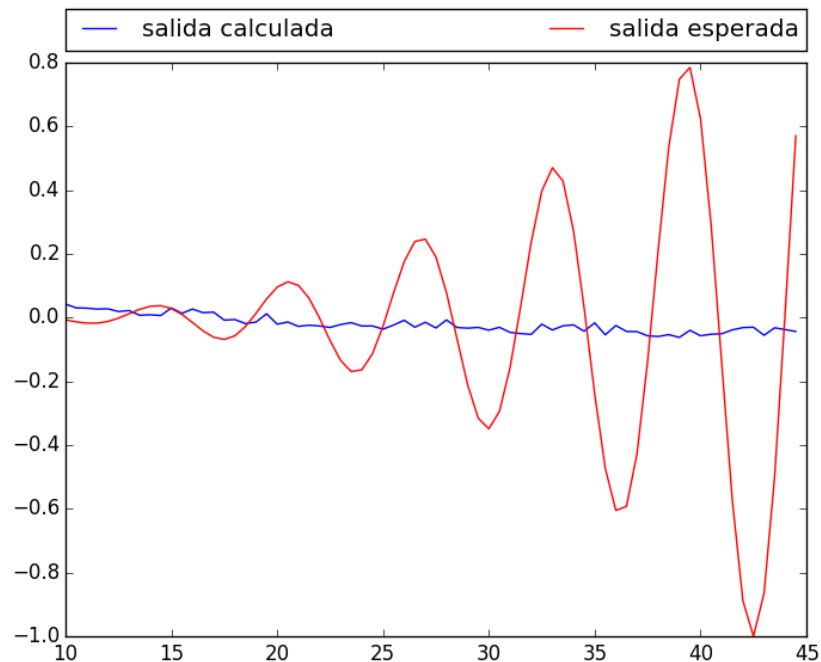
Gráfico Variante 2



η (eta) adaptativo y *momentum*

Variante:

- 350 neuronas en la capa oculta
- η : 3×10^5 - adaptativo
- g : tanh
- rango = (10, 45, 0.5)
- *momentum*



Variantes del Dominio de $f(x)$

Para lograr una mejor aproximación, se decidió cambiar el dominio de la función a estimar. El dominio elegido fue $[10, 15]$, con intervalo variante.

- Variante 1:
 - arq: 25-10, $\eta = 0.05$
- Variante 2:
 - arq: 25-10, $\eta = 0.0005$, con *momentum*
- Variante 3:
 - arq: 32-16-16, $\eta = 0.05$, $g = \exp$
- Variante 4:
 - arq: 100, $\eta = 0.005$

Gráfico Variante 1

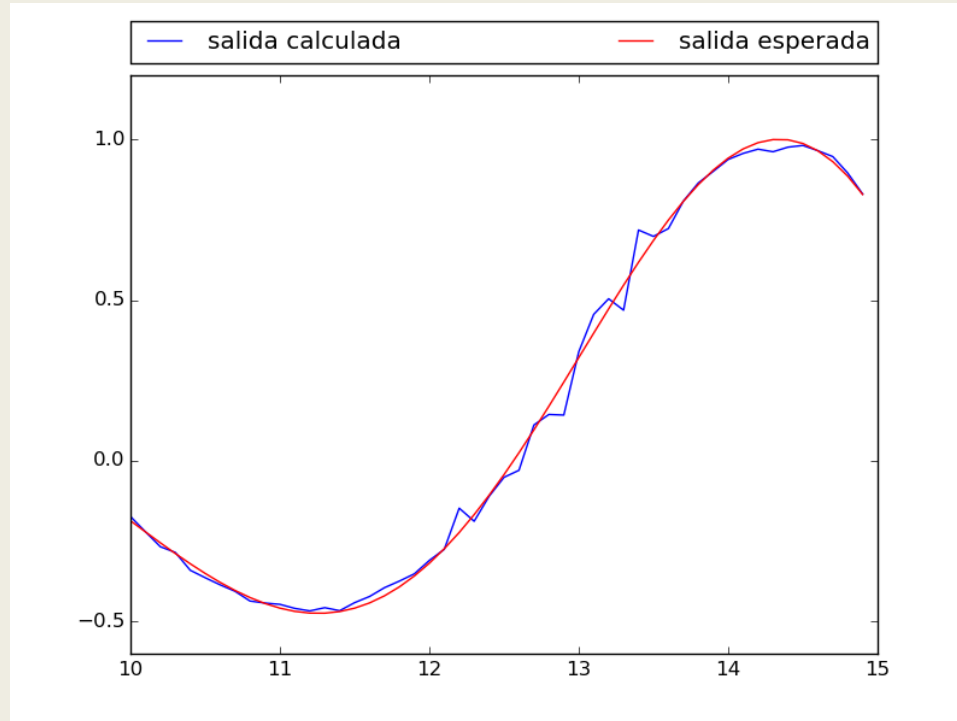


Gráfico Variante 2

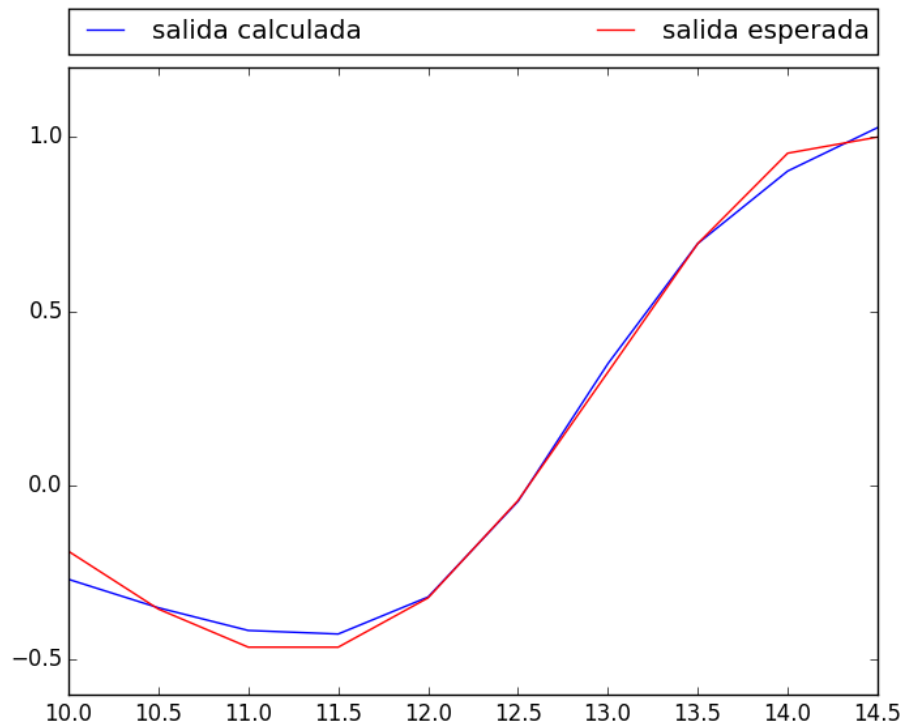


Gráfico Variante 3

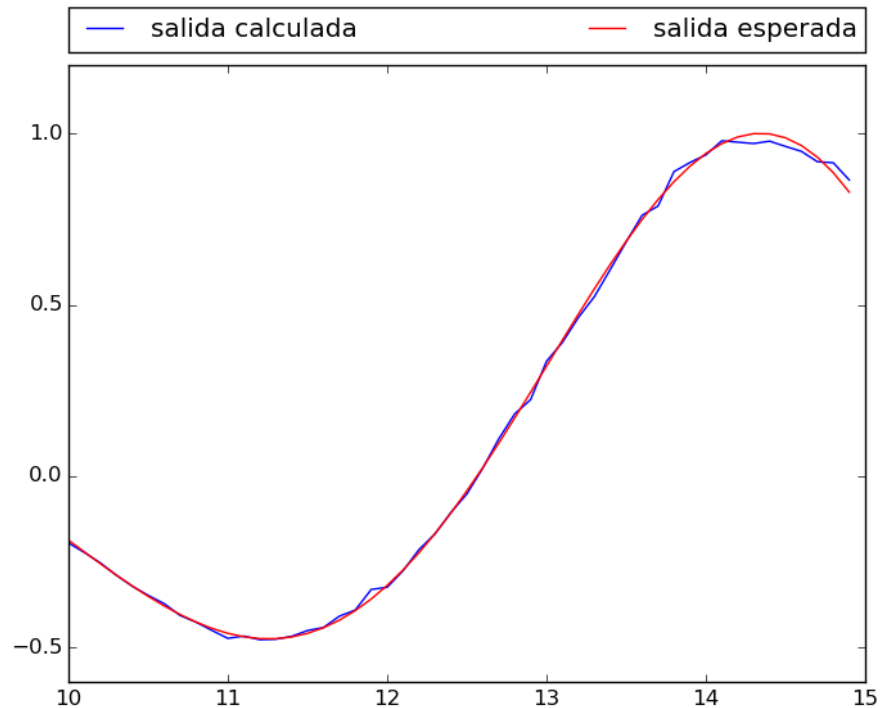
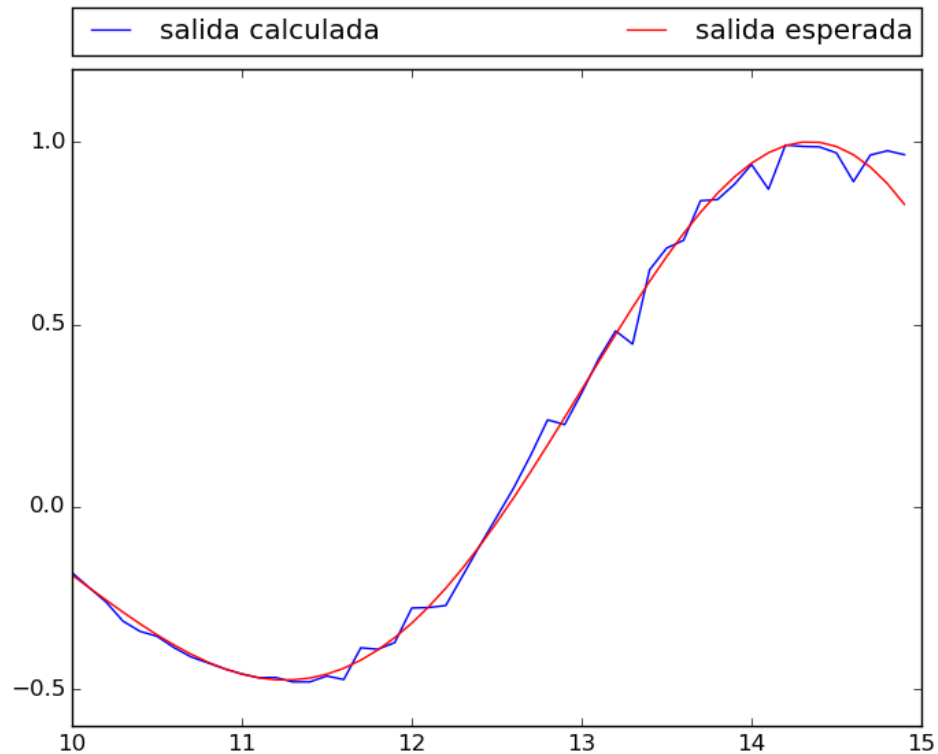


Gráfico Variante 4



Mejoras implementadas

- η adaptativo
- Momentum
- Acotar la inicialización de los pesos

Resultados comparativos

Arquitectura	η	fn	α	β	α_m	épocas	E_{ent}	$E_{gen(0,001)}$
100	5×10^{-2}	<i>exp</i>	0	0	0	50000	$3,34 \times 10^{-3}$	$3,40 \times 10^{-2}$
100	5×10^{-2}	<i>tan</i>	0	0	0	50000	$1,62 \times 10^{-3}$	$1,74 \times 10^{-2}$
25 – 10	5×10^{-2}	<i>exp</i>	0	0	0	50000	$9,91 \times 10^{-3}$	$9,94 \times 10^{-3}$
25 – 10	5×10^{-2}	<i>tanh</i>	0	0	0	8730	$9,33 \times 10^{-5}$	$9,39 \times 10^{-5}$
32 – 16 – 16	5×10^{-2}	<i>exp</i>	0	0	0	50000	$2,1 \times 10^{-4}$	$2,15 \times 10^{-4}$
32 – 16 – 16	5×10^{-2}	<i>tanh</i>	0	0	0	4915	$8,41 \times 10^{-5}$	$8,49 \times 10^{-5}$
32 – 16 – 16	5×10^{-2}	<i>tanh</i>	0	0	0,3	3560	$9,82 \times 10^{-5}$	$8,91 \times 10^{-5}$
32 – 16 – 16	5×10^{-2}	<i>tanh</i>	4×10^{-3}	10^{-1}	0	3438	$9,90 \times 10^{-5}$	$9,98 \times 10^{-5}$
32 – 16 – 16	5×10^{-2}	<i>tanh</i>	4×10^{-3}	10^{-1}	0,3	3244	$8,9 \times 10^{-5}$	$9,12 \times 10^{-5}$

Conclusiones

- Las arquitecturas de varias capas fueron, en general, superiores a las arquitecturas de una sola capa
- La función de activación tanh resultó ser más efectiva que la función exponencial
- Utilizando *momentum*, se logró obtener un error menor, en menor cantidad de épocas
- Utilizando η adaptativo, se observó el mismo efecto que con la mejora de *momentum*